

### *AMENDMENT TO THE CLAIMS*

1. *(Currently Amended)* A voter for a redundant system with n modules wherein each of said n modules generates a word output, comprising:
  - (a) a word voter receiving said word output from each of said n modules, wherein said word voter comprises inexact or threshold matching; and
  - (b) a voter decision generated by said word voter utilizesing a word basis of said word output of each of said n modules.
2. *(Original)* The voter as set forth in claim 1, wherein said word voter is based on a majority voting principle.
3. *(Original)* The voter as set forth in claim 1, wherein said word output comprises two or more bits.
4. *(Original)* The voter as set forth in claim 1, wherein said word voter comprises  $\left( \begin{matrix} n \\ \lceil n/2 \rceil \end{matrix} \right)$  matching circuits.
5. *(Original)* The voter as set forth in claim 1, wherein said word voter comprises exact matching.
6. *(Canceled)*

7.    (*Original*) The voter as set forth in claim 1, further comprising an error signal when not a majority of modules in said n modules generates a same word output.
8.    (*Original*) The voter as set forth in claim 7, wherein said error signal is a self-checking error signal.
9.    (*Original*) The voter as set forth in claim 1, wherein an entire word of said word output is considered by said word voter.
10.   (*Original*) The voter as set forth in claim 1, wherein a subset of each word in said word output is considered by said word voter.
11.   (*Original*) The voter as set forth in claim 1, wherein said redundant system is a circuitry system.
12.   (*Original*) The voter as set forth in claim 11, wherein said circuitry system comprises a logical circuitry module.
13.   (*Original*) The voter as set forth in claim 1, wherein said redundant system comprises a data or signal processing module.

14. *(Original)* The voter as set forth in claim 1, wherein said redundant system is a triple modular redundancy system and said n equals 3.
15. *(Original)* The voter as set forth in claim 1, wherein said redundant system is a triple modular redundancy simplex system and said n equals 3 and configures to a simplex system comprising a non-faulty module wherein said n equals 1 in case one of said n modules becomes faulty.
16. *(Original)* The voter as set forth in claim 1, wherein said n modules have different implementations with a same functionality.
17. *(Original)* The voter as set forth in claim 1, wherein said n modules have same implementations with a same functionality.
18. *(Currently Amended)* A method of designing circuitry systems, comprising the steps of:
  - (a) providing a redundant system with n modules wherein each of said n modules generates a word output;
  - (b) providing a word voter receiving said word output from each of said n modules, wherein said word voter comprises inexact or threshold matching; and
  - (c) generating a voter decision by said word voter ~~utilizesing~~ a word basis of said word output of each of said n modules.

19. (*Original*) The method as set forth in claim 18, wherein said word voter is based on a majority voting principle.
20. (*Original*) The method as set forth in claim 18, wherein said word output comprises two or more bits.
21. (*Original*) The method as set forth in claim 18, wherein said word voter comprises  $\binom{n}{\lceil n/2 \rceil}$  matching circuits.
22. (*Original*) The method as set forth in claim 18, wherein said word voter comprises exact matching.
23. (*Canceled*)
24. (*Original*) The method as set forth in claim 18, further comprising the step of generating an error signal when not a majority of modules in said n modules generates a same word output.
25. (*Original*) The method as set forth in claim 24, wherein said error signal is a self-checking error signal.

26. *(Original)* The method as set forth in claim 18, wherein an entire word of said word output is considered by said word voter.
27. *(Original)* The method as set forth in claim 18, wherein a subset of each word in said word output is considered by said word voter.
28. *(Original)* The method as set forth in claim 18, wherein said redundant system is a circuitry system.
29. *(Original)* The method as set forth in claim 28, wherein said circuitry system comprises a logical circuitry module.
30. *(Original)* The method as set forth in claim 18, wherein said redundant system comprises a data or signal processing module.
31. *(Original)* The method as set forth in claim 18, wherein said redundant system is a triple modular redundancy system and said n equals 3.
32. *(Original)* The method as set forth in claim 18, wherein said redundant system is a triple modular redundancy simplex system and said n equals 3 and configures to a simplex system comprising a non-faulty module wherein said n equals 1 in case one of said n modules becomes faulty.

33. (*Original*) The method as set forth in claim 18, wherein said n modules have different implementations with a same functionality.
34. (*Original*) The method as set forth in claim 18, wherein said n modules have same implementations with a same functionality.
35. (*Currently Amended*) A method of using a word voter for hardware systems, comprising the step of:
- (a) providing a redundant system with n modules wherein each of said n modules generates a word output;
  - (b) providing a word voter receiving said word output from each of said n modules, wherein said word voter comprises inexact or threshold matching; and
  - (c) generating a voter decision by said word voter ~~utilizesing~~ a word basis of said word output of each of said n modules.
36. (*Original*) The method as set forth in claim 35, wherein said word voter is based on a majority voting principle.
37. (*Original*) The method as set forth in claim 35, wherein said word output comprises two or more bits.

38. *(Original)* The method as set forth in claim 35, wherein said word voter comprises  $\binom{n}{\lceil n/2 \rceil}$  matching circuits.
39. *(Original)* The method as set forth in claim 35, wherein said word voter comprises exact matching.
40. *(Canceled)*
41. *(Original)* The method as set forth in claim 35, further comprising the step of generating an error signal when not a majority of modules in said n modules generates a same word output.
42. *(Original)* The method as set forth in claim 41, wherein said error signal is a self-checking error signal.
43. *(Original)* The method as set forth in claim 35, wherein an entire word of said word output is considered by said word voter.
44. *(Original)* The method as set forth in claim 35, wherein a subset of each word in said word output is considered by said word voter.

45. *(Original)* The method as set forth in claim 35, wherein said redundant system is a circuitry system.
46. *(Original)* The method as set forth in claim 45, wherein said circuitry system comprises a logical circuitry module.
47. *(Original)* The method as set forth in claim 35, wherein said redundant system comprises a data or signal processing module.
48. *(Original)* The method as set forth in claim 35, wherein said redundant system is a triple modular redundancy system and said  $n$  equals 3.
49. *(Original)* The method as set forth in claim 35, wherein said redundant system is a triple modular redundancy simplex system and said  $n$  equals 3 and configures to a simplex system comprising a non-faulty module wherein said  $n$  equals 1 in case one of said  $n$  modules becomes faulty.
50. *(Original)* The method as set forth in claim 35, wherein said  $n$  modules have different implementations with a same functionality.
51. *(Original)* The method as set forth in claim 35, wherein said  $n$  modules have same implementations with a same functionality.



52. *(Original)* A word voter with  $n$  outputs for a triple modular redundancy system, comprising:
- (a) three pair-wise matching circuits each having  $n$  XNOR gates and  $n$  2-input AND gates to receive output vectors of modules of said triple modular redundancy system; and
  - (b)  $2n$  2-input AND gates and  $n$  2-input OR gates to generate said  $n$  outputs.
53. *(Original)* The word voter as set forth in claim 52, further comprising a one 3-input NOR gate to generate an error signal independent of said  $n$  outputs.
54. *(Original)* A word voter with  $n$  outputs for a triple modular redundancy simplex system comprising:
- (a) three pair-wise matching circuits each having  $n$  XNOR gates and  $n$  2-input AND gates to receive output vectors of modules of said triple modular redundancy simplex system;
  - (b)  $2n$  2-input AND gates and  $n$  2-input OR gates to generate said  $n$  outputs;
  - (c) five 2-input AND gates and one 2-input OR gate independent of said  $n$ ; and
  - (d) three flip-flops independent of said  $n$ .
55. *(Original)* The word voter of claim 54, further comprising a one 3-input NOR gate to generate an error signal independent of said  $n$ .

56. (Original) A word-voter with  $n$  outputs for a  $N$  modular redundancy system comprising:

(a)  $\binom{N}{\lceil N/2 \rceil}$  matching circuits; and

(b) said matching circuits having  $n$  sub-circuits each comprising two  $\lceil N/2 \rceil$ -input AND gates and one 2-input OR gate.

57. (Original) The word voter of claim 56, further comprising one  $\binom{N}{\lceil N/2 \rceil}$ -input NOR gate to generate an error signal independent of said  $n$ .

58. (Original) A word-voter with  $n$  outputs for a  $N$  modular redundancy system comprising:

(a)  $\binom{N}{\lceil N/2 \rceil}$  matching circuits;

(b) said matching circuit having a minimum number of pair-wise matching circuits, wherein said minimum number is derived for each said  $N$  using a covering procedure;

(c) said pair-wise matching circuits each having  $n$  XNOR gates and  $n$  2-input AND gates to receive output vectors of modules of said  $N$  modular redundancy system; and

(d) AND gates to combine said output vectors of said pair-wise matching circuits.

59. (*Original*) The word voter of claim 58, further comprising one  $\binom{N}{\lceil N/2 \rceil}$ -input

NOR gate to generate an error signal independent of said n.